11-2021

HandyPose and VehiPose: Pose Estimation of Flexible and Rigid Objects

Divyansh Gupta
dg9679@rit.edu

HandyPose and VehiPose: Pose Estimation of Flexible and Rigid Objects

Divyansh Gupta

HandyPose and VehiPose: Pose Estimation of Flexible and Rigid Objects

Divyansh Gupta

Nov 2021

A Thesis Submitted
in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in
Computer Engineering

RIT | Kate Gleason College of Engineering

Department of Computer Engineering

HandyPose and VehiPose: Pose Estimation of Flexible and Rigid Objects

Divyansh Gupta

Committee Approval:

Dr. Andreas Savakis, *Advisor* Date
Department of Computer Engineering

Dr. Alexander Loui, *Committee Member* Date
Department of Computer Engineering

Dr. Matthew Dye, *Committee Member* Date
Department of Liberal Studies, National Technical Institute for the Deaf

Acknowledgments

I would like to take this opportunity to express my gratitude and thank my advisor Dr. Andreas Savakis for his constant support and guidance in academics and my overall development. I would also like to thank my committee members, Dr. Alexander Loui and Dr. Matthew Dye for their support and being on my thesis committee. I am grateful for the members of the Vision and Image Processing Lab at RIT, especially Bruno Artacho, who were all so welcoming and helpful. I would also like to thank my family and friends for their love and continuous encouragement during this challenging endeavor.

Abstract

Pose estimation is an important and challenging task in computer vision. Hand pose estimation has drawn increasing attention during the past decade and has been utilized in a wide range of applications including augmented reality, virtual reality, human-computer interaction, and action recognition. Hand pose is more challenging than general human body pose estimation due to the large number of degrees of freedom and the frequent occlusions of joints. To address these challenges, we propose HandyPose, a single-pass, end-to-end trainable architecture for hand pose estimation. Adopting an encoder-decoder framework with multi-level features, our method achieves high accuracy in hand pose while maintaining manageable size complexity and modularity of the network. HandyPose takes a multi-scale approach to representing context by incorporating spatial information at various levels of the network to mitigate the loss of resolution due to pooling. Our advanced multi-level waterfall architecture leverages the efficiency of progressive cascade filtering while maintaining larger fields-of-view through the concatenation of multi-level features from different levels of the network in the waterfall module. The decoder incorporates both the waterfall and multi-scale features for the generation of accurate joint heatmaps in a single stage. Recent developments in computer vision and deep learning have achieved significant progress in human pose estimation, but little of this work has been applied to vehicle pose. We also propose VehiPose, an efficient architecture for vehicle pose estimation, based on a multi-scale deep learning approach that achieves high accuracy vehicle pose estimation while maintaining manageable network complexity and modularity. The VehiPose architecture combines an encoder-decoder architecture with a waterfall atrous convolution module for multi-scale feature representation. It incorporates contextual information across scales and performs the localization of vehicle keypoints in an end-to-end trainable network. Our HandyPose architecture has a baseline of vehipose with an improvement in performance by incorporating multi-

level features from different levels of the backbone and introducing novel multi-level modules. HandyPose and VehiPose more thoroughly leverage the image contextual information and deal with the issue of spatial loss of resolution due to successive pooling while maintaining the size complexity, modularity of the network, and preserve the spatial information at various levels of the network. Our results demonstrate state-of-the-art performance on popular datasets and show that HandyPose and VehiPose are robust and efficient architectures for hand and vehicle pose estimation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The advancements in computer vision have been utilized in a variety of tasks such as object detection, segmentation, image classification, pose estimation, and others. The increasing capacity of hardware and ongoing research in the field of deep learning have accelerated the pace of innovation in all these tasks. Convolutional Neural Networks (CNN) have proven to be remarkable at extracting features from images, videos and texts. Many deep learning and computer vision methods have adapted CNN's for multiple tasks. In this work we will be using the capabilities of CNN for the task of pose estimation.

Hands are important body parts for humans to interact with and manipulate their environment. Hand pose estimation is a valuable and challenging problem in computer vision, aiming to locate a set of coordinates on an input hand image as a set of certain parts (e.g. palm and fingers) constructing a human hand representation for multiple applications.

Human-computer interaction is a rapidly evolving field that focuses on the interaction between people and computers [2], and hands play a very crucial role in making it a comfortable and convenient interactive experience [3], [4]. Hand pose recognition is a fundamental human ability and an important yet elusive goal for computer vision

research, as the human hands are prominently agile with a high degree of freedom compared to most of the other body parts. Self-occlusion of fingers also contributes to making the task more difficult, as the structure of the hand is very delicate and complex.

Hand pose estimation includes methods for 2D hand pose [5], [6], [7] and 3D hand pose estimation [8], [9]. Hand pose is more challenging than general human body pose estimation [10], [11], [12] due to the large number of degrees of freedom in the human hand and the high degree of occlusion of joints in a monocular view of the hand. To overcome the issue of localizing joints that are occluded, methods may employ statistical and geometric models or use anchor poses to estimate the occluded joints [13], [14].

Many methods for hand pose estimation focus on 3D estimation, but obtaining the complete kinematic structure of hands in 3D space is correlated to the performance of 2D hand pose estimation. There are several methods using depth images [15], [16], wearable sensors [17], external sensor devices [18] and multiple camera views for obtaining the 3D hand pose but these methods are expensive and require additional resources for pose estimation. Due to the abundance of monocular RGB images researchers have started focusing on using them for 3D hand pose estimation [19], [8], [20], [21] [22].

In recent years, attempts have been made to leverage the 2D hand pose estimation and use it as an intermediate stage for obtaining the 3D structure of hand [23] [9] [24] [22]. Although the approach is more cost and memory-efficient, still not much focus has been given to the 2D estimation part which is an important stage for many applications. We will focus on improving the 2D hand pose estimation having multiple domain applications and achieve state-of-the-art results on two of the most prominent and widely used datasets for hand pose estimation.

Each particular set of co-ordinates in the human hand are known as a joint or

a key-point. Joints are the place where two or more bones are joined by soft tissue. They can be divided into three different groups of Fibrous, Cartilaginous, and Synovial joints in increasing order of flexibility and complexities. Fibrous joints are attached by dense fibrous-connected tissues and allow for minuscule movement. Cartilaginous joints are connected by cartilaginous tissues which allow for limited amount of movement. Synovial joints are the most complex ones and allow for delicate and rapid movement. Hand joints belong to the group of Synovial joints which allow for great range of movement, making them more challenging and complex to estimate.

In general, the main approaches of pose estimation are either based on top-down methods [25], [26], and [27], or bottom-up methods [28].

In the top-down approach, we incorporate by localizing and recognizing independent object by introducing a square bounding box object detector like YOLO [29], Faster R-CNN [30] or CornerNet [31] and identify the total number of instances present in an image followed by evaluating the keypoints and estimating the position of those keypoints for every instance. These top-down methods for pose estimation are dependent on precise object detectors and can be lagging if there are multiple instances present in the image.

In the bottom-up approach, all the specific keypoints in the image are detected first, followed by clustering those keypoints belonging to several distinct instances. This offers more robustness and increases the potential to decouple runtime complexity from the total number of instances present in the image.

Hand pose estimation can be further divided into detection based [32], [11] and regression based [6], [33], [34] methods. In detection based method, the network produces a probability density map for each joint as heatmaps. The total number of heatmaps generated is proportional to the total number of joints [35]. Applying an argmax function on the corresponding heatmaps will provide the exact location of a joint in the heatmap. Whereas, in regression based method, the network tries to

directly estimate the position of each joint in an image. The total number of neurons generated in the last layer of the neural network is twice the total number of joints to predict the x and y coordinates of each joint for 2D hand pose estimation [36].

Hand Pose Estimation has drawn increasing attention during the past decade and has been utilized in a wide range of applications including augmented reality, virtual reality, human-computer interaction, aerial handwriting, and action recognition. Hand pose estimation has been closely related to human pose estimation using backbones like [11], [10] but none of the methods relate it to the task of semantic segmentation as in [37], [38], and [1]. In recent years, coupling the task of human pose estimation and semantic segmentation has been achieving state-of-the-art results for human pose estimation [12], and we successfully leveraged its benefits for the task of hand pose estimation.

Leveraging on recent advances of multi-scale feature representations with application to human pose estimation in [12] and [39], we propose HandyPose, a single-stage network for hand pose estimation that is end to end trainable and produces state-of-the-art results. To deal with the challenges of hand pose context and resolution, our architecture generates improved multi-scale and multi-level representations by combining features from multiple levels of the backbone network via our Multi-Level WASP (MLW) module.

Examples of hand pose estimation obtained with HandyPose are shown in Figure 1.1. A main component of our HandyPose architecture is the integration of Multi-Level Features (MLF) along with the extended Field-of-View (FOV) extracted by the advanced Waterfall Atrous Spatial Pooling (WASPv2) module [39]. The HandyPose encoder-decoder architecture with our multi-level waterfall module combines the cascaded approach for atrous convolutions with larger FOV with feature extraction from multiple levels of the backbone and has the potential for wider application to other tasks beyond hand pose estimation.

Figure 1.1: Pose estimation examples with our HandyPose method.

HandyPose predicts the location of hand joints by utilizing contextual representations obtained with the multi-scale and multi-level scheme taken in our network. Our contextual representation approach allows better detection of shapes, resulting in a more accurate estimation of occluded joints, without requiring postprocessing relying on statistical or geometric methods.

Vehicle pose estimation is also an important task having multiple applications but has not been explored much compared to human pose estimation. With the recent advancements in technology for the automotive industry, the demand for accurate vehicle pose estimation [40] has gained popularity due to its applications in autonomous driving [41], traffic monitoring [42] and scene analysis. Vehicle pose estimation involves locating specific keypoints of a particular vehicle under consideration. This is a challenging task, as there are several types of vehicles with different color, shape,

and size.

CNN's have revolutionized the field of deep learning and have been used to dramatically improve the performance of human pose estimation methods [28]. However, very little of these methods has been utilized in vehicle pose estimation. Human pose estimation is challenging due to high degree of freedom in body joints and high occlusion of those joints, whereas vehicle pose deals with a more rigid structure and has different types of occlusions.

The growth of the automobile industry has resulted in high variability within each vehicle class, causing challenges for developing a reliable method for different types of vehicles. Camera viewpoint has more variations in elevation for vehicles. So far, vehicle datasets [43] are annotated for other tasks and there are no defined conventions for pose, making it difficult to find representative keypoints for training and testing deep learning models.

To deal with above challenges and improve on the generalization power of the network, our framework utilizes an encoder-decoder architecture that leverages multi-level features from the backbone (ResNet-101) and processes them with a waterfall module [1] for multi-scale representations. A related version of this configuration, without multi-level features, was beneficial for the tasks of semantic segmentation [1] and human pose estimation [12]. In this thesis work, we incorporate multi-level features in the waterfall module and demonstrate the usefulness of our framework for vehicle pose estimation.

Our architecture combines an encoder-decoder network along with larger field of view generated by the waterfall of atrous convolutions. Aiming to achieve better spatial and contextual representations, our multi-scale approach is designed to improve the predicted keypoint accuracy by combining atrous convolutions and low-level feature maps from the encoder network, and integrating them with the decoder module. This approach generates richer image features by concatenating them and avoiding

loss of spatial information at different scales.

The multi-scale approach, along with successively increasing the Field-of-View (FOV) in a waterfall architecture, helps in predicting the location of keypoints by preserving the contextual and spatial information. Our approach more efficiently incorporates the contextual information across scales and performs keypoint localization in a single stage, end-to-end trainable network. Our results demonstrate that VehiPose is a robust and efficient architecture for vehicle pose estimation.

The main contributions of this thesis are the following.

- We propose the novel HandyPose framework, a multi-level and multi-scale, end-to-end trainable, single-stage approach that produces state-of-the-art results for hand pose estimation.

- We propose the improved multi-level waterfall module for feature extraction that more effectively encodes feature maps while maintaining high resolution, without significantly increasing the computational cost or the size of the network.

- The proposed HandyPose framework is based on an encoder-decoder architecture and incorporates multi-level features in both the encoder and the decoder. The proposed architecture is easy to modify and expand for application to a wide variety of tasks, due to its modularity and capacity for extracting contextual information from the feature extractor.

- We propose the VehiPose framework, a multi-scale, end-to-end trainable, single-stage approach that produces state-of-the-art results for vehicle pose estimation.

- The waterfall framework generates multi-scale feature representations by combining the contextual and spatial information, resulting in larger FOV features for vehicle pose estimation.

The remainder of this proposal is organized as follows: Chapter 2 discusses the related work and background knowledge in the fields of hand and vehicle pose estimation; Chapter 3 describes our vehicle pose estimation research including the approach used, experiments performed and the results obtained; Chapter 4 describes the approaches used in the development of state-of-the-art hand pose estimation method proposed in this research; Chapter 5 discusses the detailed description of the experiments performed on various hand datasets; Chapter 6 presents the state-of-the-art results obtained on various datasets. Finally, Chapter 7 demonstrates the conclusion of this thesis.

Chapter 2

Background

2.1 Related Work

Before the meteoric growth of deep learning methods, hand pose estimation was a laborious task that was rarely deployed in applications due to the setup expense and algorithm complexity. Mostly traditional computer vision algorithms were applied for estimating the hand pose. Early methods include k-nearest neighbors and decision trees [44]. For instance, [45] applied a multi-colored glove for reconstructing the pose of a hand from a single image, and relied on the nearest neighbors technique to map the hand detection using the multiple colors of the glove. Marker-based methods were expanded by [46] and [47], using a setup with multiple cameras. A downside of these techniques is the high cost of implementation, complex setup, and requirement of multiple cameras.

Aiming to reduce the setup complexity, [48] combines the use of a single camera setup with Bayesian technique to better connect and predict the hand pose estimation. Despite the reduced complexity, this method still relies on the color pattern of gloves to infer the pose of the hand. The Microsoft Kinect sensor [49] introduced a less complex setup with the use of a single camera and the addition of a depth sensor to extract the 3D hand pose estimation. These methods relied on random forests [44] to assess the hand rotation.

Random Forest [44] and its variations were the most successful techniques used for hand pose estimation at that time. Microsoft launched the Kinect cameras [49] with the purpose of eliminating game controllers from Microsoft Xbox 360, by using a depth-sensing camera and random forest as a classifier for human pose estimation and making its way in the commercial market by selling more than 10 million units within few months of launch.

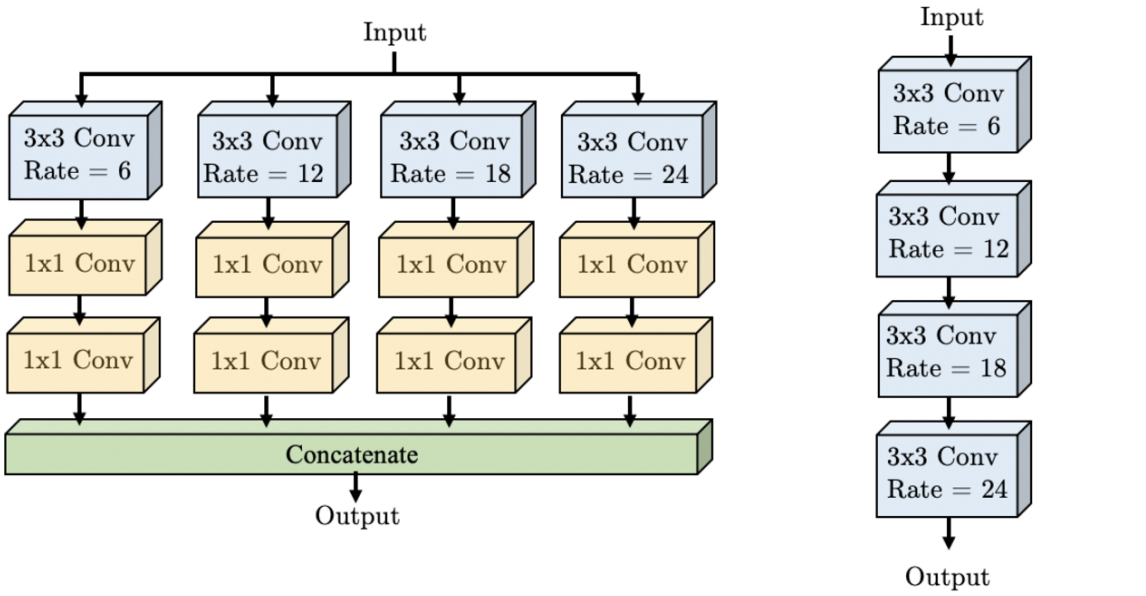

Figure 2.1: Comparison for ASPP and cascade configuration.

Firstly, they normalized the depth map data using the nearest neighbors value to be invariant to rotation. Then, they labeled each part of the body (similarly for hands) with a label and tried to classify each pixel as one of these labels [50].

2.1.1 Deep Learning Methods

More recently, deep learning methods achieved more accurate hand pose estimation and quickly gained in popularity [5], [7], [51], [52]. Some methods combine depth

estimation to extract the full 3D coordinates in addition to the 2D detections [53], [54], [35], [55]. Another approach employs Recurrent Neural Networks (RNNs) to extract spatial information of the joints and palm [56]. In the medical imaging domain, single hand X-ray images were used for hand pose estimation in [57].

Many methods follow a two-stage approach, i.e., first estimate the 2D pose of hand and then uplift it to obtain the 3D hand pose estimation. Tompson et al. [58] generated the 2D heatmaps to infer the 3D hand pose using a CNN based model along with inverse kinematics. Liu et al. [59] used an hourglass network along with 2D spatial information to estimate 3D hand pose using 2D joint detections and depth regression. Surprisingly, not much focus has been given to accurately estimating the 2D hand pose using a simple RGB image which can be the baseline for 3D hand pose estimation.

The similarity of the hand pose estimation task to human pose estimation, allows the adoption of methods developed for the overall human body. The Convolutional Pose Machine (CPM) approach [11] is popular for human pose estimation due to its easy implementation and the modularity of joint detections via the refinement of feature maps through multiple stages of the network. CPM was later expanded to integrate the concept of Part Affinity Fields (PAF) resulting in the widely used OpenPose method [28]. Leveraging on the innovations of CPM, the approach in [32] developed a multi-view bootstrapping method that implements a CPM-based architecture for 2D hand pose estimation, relying on a detector for generating a large dataset of hand keypoints. Similarly, the Hourglass (HG) network [10] stacks up to 8 iterations of its network to refine feature maps.

Pose estimation methods may be categorized as top-down [25], [26], [27] or bottom-up [28], [60]. Top-down approaches rely on an object detection stage to locate instances of the person or pose by using detectors such as YOLO [29] or Faster R-CNN [30]. The detection stage is followed by the detection of keypoints to estimate pose

for every instance. The High-Resolution Network (HRNet) [61] combines multi resolutions throughout the network, while also maintaining high-resolution feature maps through all layers of the network. Despite achieving high accuracy for individual poses, top-down methods are dependent on the performance of object detectors.

Bottom-up approaches initially detect all keypoints in the image, followed by keypoint clustering for pose estimation of separate instances. Top-Down methods can be expanded to Bottom-Up approaches by incorporating offset regression into their decoder, for instance the HigherHRNet [60] leverages the promising results of the HRNet method to achieve high accuracy bottom-up pose estimation. Both top-down [32] and bottom-up [6], [33], [34] approaches can be utilized for hand pose estimation.

Santavas et al. [6] proposed the AttentionNet for hand pose by combining a self-attention module [62] with a feed-forward CNN for directly estimating the hand keypoints without intermediate supervision. AttentionNet adopts a regression based approach instead of pixel-wise classification, which results in high processing speed but is limited in generalization performance.

Aiming to achieve a better structural learning, a non-parametric structure regularization approach [63] utilizes the synthetic hand mask for learning the structure of keypoints. Mask-Pose [5] uses the silhouette information and builds a two-stage cascade network that includes a mask prediction stage and a pose prediction stage. The SRHandNet [7] approach regresses hand regions of interest (ROI) simultaneously with hand keypoints to improve the performance of hand pose estimation.

MediaPipe [64] by Google has also made considerable advancements in object tracking and pose estimation. MediaPipe Hands [65] recently introduced multi-hand detection, the main component in it is regarding the palm detector which can detect both palms and then run a regression model on both of them to predict the coordinates. The BlazePalm detector, which is inspired from the face detection model

BlazeFace, provides a score/threshold value to estimate the presence of a hand in the image, binary classification (Left or Right hand) and the bounding box coordinates. This work is aiming for real-time applications, and to reduce the computational cost, the detector is not applied on every frame and only detects the first frame of the hand or if the hand is lost from the frame. Otherwise the bounding box is detected from the landmark predictions of the previous frame.

2.1.2 Graphical Methods

Many methods apply techniques to assess the geometric constraints of hand joints [66]. One approach is to employ graphical methods to better understand the articulation of hand joints [67]. The Adaptive Graphical Model Network (AGMN) [51] aims to learn adaptive parameters for the graphical model for each input image and refine its pose estimation. In another approach, SIA-GCN [68] uses a modified graph neural network for hand pose estimation, representing features at each node by a 2D spatial confidence map instead of 1D vectors, with the goal of preserving the spatial information provided in the 2D feature maps.

Using graphical models along with a CNN, R-MGMN [52] associated the spatial relationships with input hand shape in order to reduce the spatial irregularity of hand keypoints. More recently, the Hand-Object Pose Estimation Network (HOPE-Net) [69] applies graph convolutions in a modified U-Net configuration [70] to improve the hand pose estimation in conjunction with an object that is picked by the hand.

2.1.3 Multi-Scale Feature Representations

A challenge faced by networks using CNNs for pose estimation is the significant reduction of the resolution caused by the repeated use of pooling layers. For semantic segmentation, Deconvolution Networks [71] employed deconvolution layers to address the problem, by upsampling the resolution of each layer in the decoder stage of the

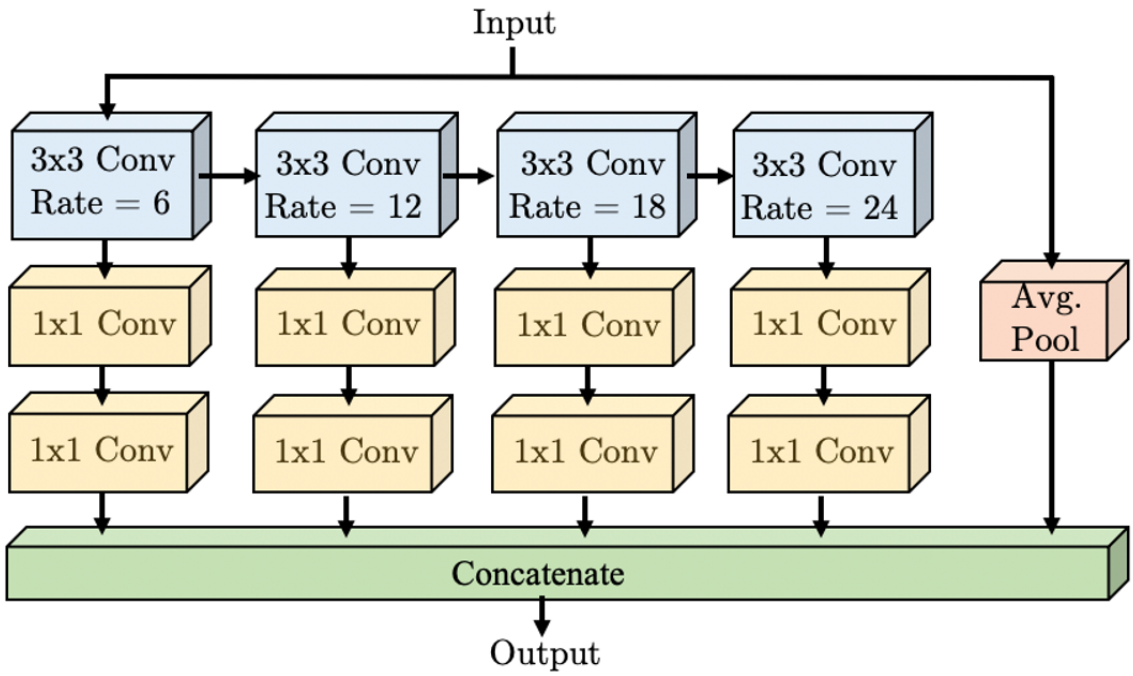

Figure 2.2: Waterfall architecture in the WASP module [1].

network.

Other methods rely on the use of atrous convolutions to avoid downsampling and increase the size of the receptive fields in the network. Atrous convolutions are applied systematically by using a multi-scale context aggregation module [72] in order to better preserve the contextual information of the input image.

Deeplab [38] further explored the advantages of atrous convolution by proposing the Atrous Spatial Pyramid Pooling (ASPP) module for semantic segmentation as shown in Figure 2.1. The ASPP approach increases the FOV at larger dilation rates in parallel branches without downsampling. The main challenge this network faces is the increased computational cost and memory requirements due to its increased resolution.

DeepLabv3 [73] addressed this issue by applying a cascade of atrous convolutions in a sequential order with the help of progressive filtering to maintain the FOV at different layers of the network. In another formulation, Res2Net [74] used a multi-

scale approach for feature extraction by introducing hierarchical connections in a single residual block of the CNN model. The Res2Net block can be plugged into many CNN based models for multi-scale feature extraction.

The Waterfall Atrous Spatial Pyramid (WASP) module [1] was initially proposed for semantic segmentation and was used in UniPose [12] for human pose estimation. The WASP module as shown in Figure 2.2 operates in a waterfall-like flow, progressively extracting the larger FOV from a series of atrous convolutions at different dilation rates. The waterfall architecture effectively generates multi-scale features from the backbone without immediately parallelizing the input stream, as it maintains the advantages of the ASPP module with lower computational and memory requirements. It incorporates the multi-scale represenations of the Res2Net block. The WASP module goes beyond the cascade approach by combining the streams from all its branches and average pooling of the original input to achieve a multi-scale representation.

The waterfall approach was recently enhanced by the introduction of the WASPv2 module for multi-person pose estimation in OmniPose [39]. The WASPv2 architecture extracts feature maps at multiple scales, while preserving the original resolution by avoiding downsampling. The cascade approach used in WASPv2 maintains the high resolution of the feature maps by arranging atrous convolutions in a cascaded structure with increasing dilation rates of [1,6,12,18]. This arrangement increases the FOV in the feature representations without affecting the input resolution.

The WASP module outputs 256 feature maps which get concatenated with the 48 high level feature maps as an output for the decoder module whereas in WASPv2 the decoder module is inbuild and it produces k feature maps as output where k is the output number of joints in an image of the dataset.

Furthermore, WASPv2 integrates the decoder with the feature extraction process, effectively reducing the overall computational cost. This effective multi-scale architecture of WASPv2 achieves high accuracy for human pose estimation. In this thesis

work, we adopt the waterfall architecture for hand pose estimation and extend it with multi-level features, integrating the atrous convolutions with the decoder module for improving the accuracy and achieving state-of-the-art results for 2D human pose estimation.

2.1.4 Top-down and Bottom-up approaches

Vehicle pose estimation is a relatively new topic with multiple applications, such as traffic surveillance and autonomous driving. However, there are very few methods for estimating the vehicle pose. There are essentially two main approaches to pose estimation: the top-down approach as shown in [26], [32] and the bottom-up approach as shown in [27], [75], and [60]. The top-down [76] approaches begin by detecting and localizing objects independently, using a bounding box object detector, such as YOLO [77] or Faster R-CNN [30]. After identifying the total number of instances present in the image, the locations of the keypoints are estimated for every instance. These top-down methods for pose estimation are dependent on precise object detection and suffer if the object detector fails. Furthermore, their runtime is directly proportional to the total number of people in the image, webcam or video, for each person detection, a single-person pose estimator is run which is very time consuming.

In a bottom-up approach [40], all the keypoints in the image are detected first, followed by clustering those keypoints belonging to distinct instances. The bottom-up approaches offer robustness and have the potential to decouple runtime complexity from the total number of instances present in the image.

Stacked hourglass networks [10] were proposed for human pose estimation and have been utilized for vehicle pose estimation in Ref. [78], [79], and [80]. These networks consist of multiple stages that are made up of residual convolutional blocks with skip connections in a symmetric design capturing information at every block.

The challenge of using an encoder feature generation module is the loss of reso-

lution due to successive pooling layers. To tackle this problem, Fully Convolutional Networks (FCN) network [81] applied upsampling techniques to upsample the image to its input dimensions. Corrales *et al.* [40] explored estimating the 2D vehicle pose in a manner similar to human pose, by proposing a simple baseline method. A ResNet [82] backbone network was utilized along with few deconvolution layers to generate heatmaps corresponding to vehicle keypoints. This approached obtained good results but was limited by the loss of spatial and contextual information of the input image during progressive convolutional layers in the network. Wang *et al.* estimated the vehicle keypoints for the task of vehicle re-identification, improving the performance of their model in distinguishing between similar vehicles.

2.1.5 Feature Representations with Atrous Convolution, ASPP and Res2Net

Atrous or dilated convolutions are used to increase the size of the receptive field, while maintaining the input size, and avoid the loss of resolution due to downsampling. Yu *et al.* systematically usds dilated convolutions for preserving the contextual information of the input image by proposing a multi-scale context aggregation module [72].

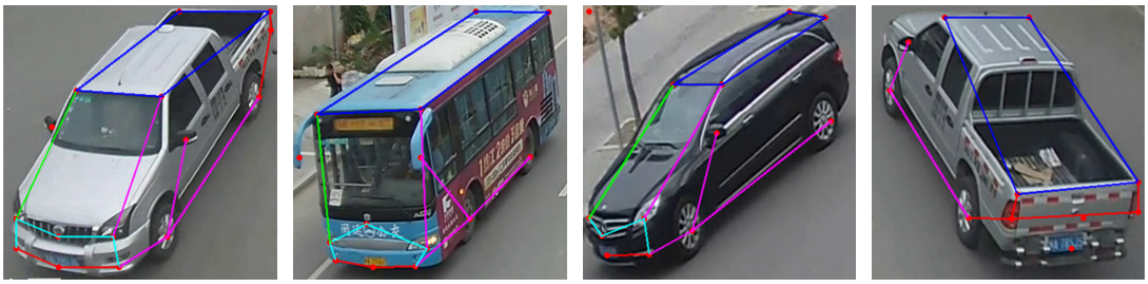

Figure 2.3: Pose estimation examples with our VehiPose method.

Further improving the FOV while maintaining the same resolution by using atrous convolutions at larger dilation rates in parallel branches, Deeplab [38] proposed the Atrous Spatial Pyramid Pooling (ASPP) module to increase the receptive field of the

network at the same resolution. DeepLab combined four branches with increasing dilation rates for larger FOV to deal with loss of resolution in the encoder module. The main disadvantage of this network was the increased computational cost and memory consumption. Res2Net [74] used a multi-scale approach for extracting features by introducing hierarchical connections in a single residual block of the CNN model. The proposed Res2Net block can be plugged into many CNN based models for multi-scale feature extraction.

Improving upon DeepLab and Res2Net, Artacho introduced the waterfall architecture of the WASP module [1] which incorporates multi-scale features of the Res2Net block and the cascade of atrous convolutions from the DeepLab model but without immediately parallelizing the input stream. The WASP module resembles a waterfall flow by progressively extracting the larger FOV from a series of atrous convolutions at different dilation rates and parallelizing the branches of the atrous convolutions. The waterfall architecture was found to be more computationally efficient and produced better results for semantic segmentation [1] and human pose estimation [12].

Chapter 3

VehiPose Vehicle Pose Estimation

3.1 VehiPose Architecture

We propose the *VehiPose* framework, a unified multi-scale framework which produces state-of-the-art results for vehicle pose estimation without any intermediate supervision or postprocessing.

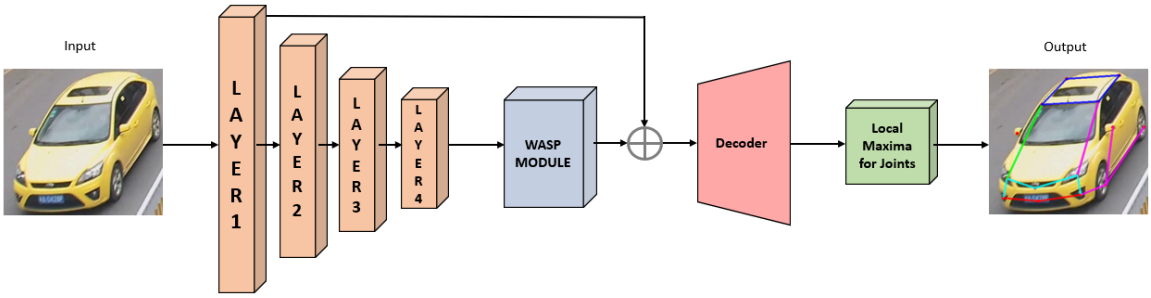

Figure 3.1: The proposed VehiPose architecture for 2D vehicle pose estimation. The input color image is fed into the ResNet backbone and the last layer features are processed by the WASP module to obtain 304 feature maps after the concatenation of WASP and low level features at $\oplus$. The decoder module generates K heatmaps, one per joint, and the exact location of each keypoint is extracted by applying a local maximum operation.

The proposed architecture is shown in Figure 3.1. The input image is fed in the ResNet backbone, generating 2048 feature maps at the second last layer of the network which are fed into the WASP module. The waterfall of atrous convolutions in the WASP module helps in preserving the spatial and contextual information due to

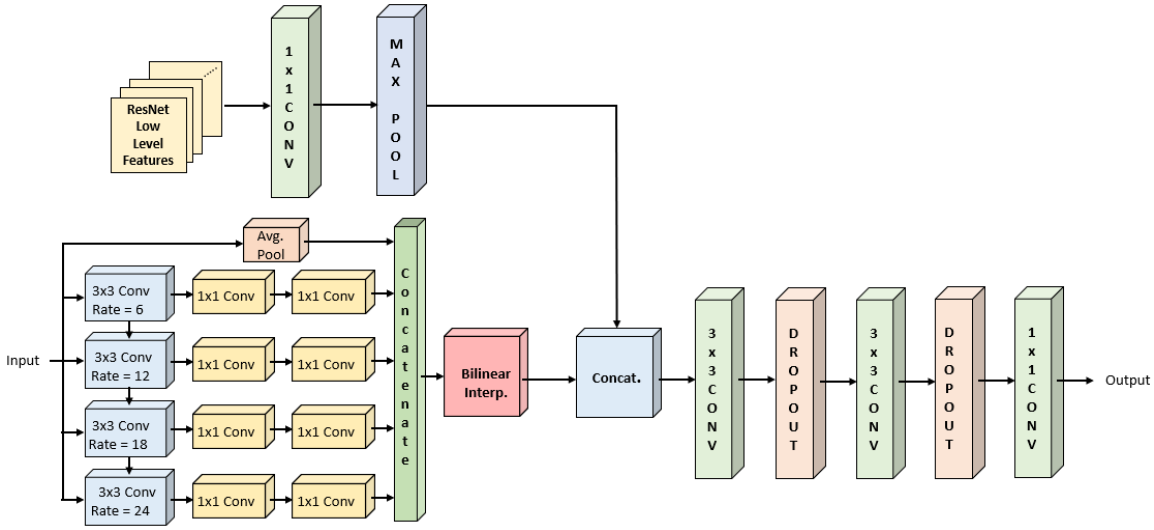

Figure 3.2: Waterfall module architecture along with the decoder module used in the VehiPose pipeline. The inputs to the decoder are 304 feature maps by concatenating 48 channels of ResNet low-level features and 256 channels of the WASP feature maps. The decoder outputs K heatmaps corresponding to K joints, where K is the total number of keypoints.

the larger Field-of-View (FOV) and multi-scale feature representation. The WASP module outputs 256 feature maps which are concatenated with 48 low-level feature maps, generated from the first block of the ResNet backbone after applying 1×1 convolution and max-pooling operation to match the dimensions. After concatenation, the 304 feature maps become the input for our decoder module, which converts the feature maps into heatmaps corresponding to the total number of keypoints.

The task of vehicle pose estimation is rigid as vehicles are not flexible, whereas hands have high degree of freedom and flexibility, hence making the task more difficult. Therefore, our HandyPose architecture shown in later chapters has a baseline of VehiPose and uses multi-level feature from the backbone to incorporate multi-level feature extraction in our proposed Multi-Level WASP and Multi-Level Decoder module.

3.1.1 WASP module

The success of atrous convolutions in the tasks of semantic segmentation [1] and human pose estimation [12] inspired us to include the waterfall of atrous convolutions in our architecture for the task of vehicle pose estimation. The proposed waterfall architecture, along with the decoder module, is shown in Fig. 3.2. The four branches in WASP have different FOV and are arranged in a waterfall-like fashion. The WASP module goes beyond the cascade approach by combining the streams from all its branches and average pooling of the original input to achieve a multi-scale representation. WASP is designed with the goal of reducing the number of parameters in order to deal with memory constraints and overcome the computational limitation of atrous convolutions.

In this thesis, we have also developed an improvement of the WASP and WASPv2 module for our HandyPose architecture utilizing the multi-level and multi-scale approach for larger field-of-view in the waterfall module as shown in later chapters.

3.1.2 Decoder module

The decoder module combines the 256 feature maps coming from the WASP module with ResNet low level feature containing 48 feature maps, forming a total of 304 feature channels. Our decoder module converts the 304 feature maps to heatmaps, each corresponding to a joint or keypoint.

The output consists of K heatmaps that are used for keypoint localization after performing a local maximum operation in each heatmap. The decoder module helps in increasing the performance of the network by combining high and low level feature maps and processing them to progressively reduce the feature maps and gradually achieve K output feature maps

Improving upon the decoder module used in the VehiPose architecture, we have developed a multi-level decoder module for our HandyPose architecture for more

thoroughly utilizing the multi-level features from the backbone and improving the results for the challenging task of hand pose estimation.

3.2 Experiments

For VehiPose, we performed experiments on the VeRi-776 dataset [83] composed of single vehicle images. VeRi-776 dataset consists of more that 50,000 images. Each image contains 20 labelled keypoints annotations for a single vehicle.

3.2.1 Datasets

3.2.2 VeRi-776 Datasets

We performed experiments on the VeRi-776 dataset [83] composed of single vehicle images. VeRi-776 dataset consists of more that 50,000 images. Each image contains 20 labelled keypoints annotations for a single vehicle. The keypoints are annotated keeping in mind the most common parts which are present in almost all vehicles like the wheels, headlights and so on.

Table 3.1 presents the details of the keypoint locations. Vehicles are mostly centrally located in images, allowing a good assessment of the network performance for the task of single vehicle pose estimation.

3.3 Evaluation Metric

For the evaluation of VehiPose, we used Percentage of Correct Keypoints (PCK) as the evaluation metric. It considers the prediction to be correct when the keypoint lies within a certain threshold σ from the ground truth location. For Example, $PCK(@0.2) = P(\sigma)/K$, means for a threshold σ of 0.02 and input image of size w $\times$ w, PCK is defined as the number of predicted keypoints (P) that are within the

Index	Location	Index	Location
1	left-front wheel	11	left rear-view mirror
2	left-back wheel	12	right rear-view mirror
3	right-front wheel	13	right-front corner of vehicle top
4	right-back wheel	14	left-front corner of vehicle top
5	right fog lamp	15	left-back corner of vehicle top
6	left fog lamp	16	right-back corner of vehicle top
7	right headlight	17	left rear lamp
8	left headlight	18	right rear lamp
9	front auto logo	19	rear auto logo
10	front license plate	20	rear license plate

Table 3.1: VeRi-776 dataset keypoint positions.

threshold range of $\sigma \times 0.2$ of the ground truth keypoints location divided by the total number of keypoints (k).

3.4 Implementation Details

For VehiPose, we considered different rates of dilation on the WASP module and larger rates resulted in better prediction. A set of dilation rates of r = 6, 12, 18, 24 was selected for the WASP module. Training was performed for 100 epochs with a batch size of 16 images. The learning rate was set initially at 10^{-4} and then reduced progressively for best results.

3.5 VehiPose Results

3.5.1 Experimental Results on VeRi-776 Dataset

We tested VehiPose on VeRi-776 dataset and obtained the results shown in Table 3.2. We performed a series of experiments to compare the performance of ASPP and WASP modules with our decoder module. We also reported the computational cost

and number of parameters of each network to show the computational complexity and memory requirements.

The WASP module performs better than ASPP, improving PCK@0.2 results by 1.75% for vehicle pose estimation. In addition, it is computationally more efficient and requires fewer parameters. Examples of VehiPose detections for the VeRi-776 dataset are shown in Figure 3.5 and 3.4. These examples illustrate that VehiPose deals effectively with occlusion and vehicles with different color, size, and shape.

ASPP	WASP	Decoder	PCK@0.2	Params (M)	GFLOPs
-	-	✓	53.15	47.8	35.5
✓	-	✓	54.37	59.3	34.9
-	✓	✓	56.12	47.5	29.2

Table 3.2: Results on VeRi-776 dataset using various configurations of the VehiPose framework with a ResNet backbone.

The output heatmaps generated by our VehiPose architecture are shown in Figure 3.3. We generate k heatmaps where k is equivalent to the total number of annotated keypoints in the single image including wheels, headlights, logo, and parts of the top and bottom of vehicle.

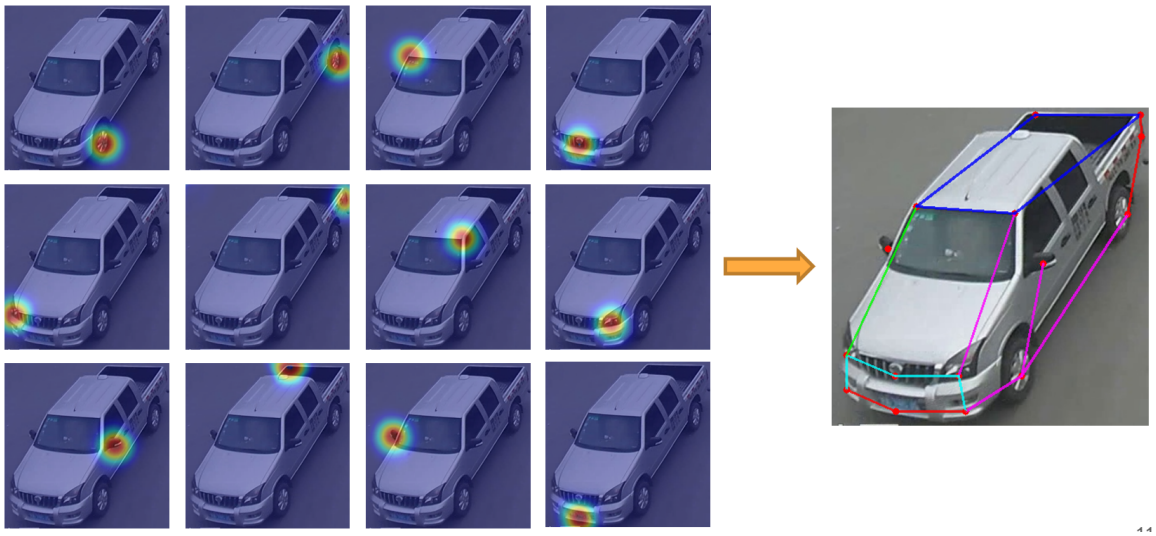

Figure 3.3: Heatmaps generated and final pose estimated from VehiPose method.

Figure 3.4: Vehicle pose estimation examples from the VeRi-776 dataset.

Figure 3.5: Vehicle pose estimation examples from the VeRi-776 dataset.

Chapter 4

HandyPose Architecture

4.1 HandyPose Architecture

We propose *HandyPose*, a multi-level framework for hand pose estimation, that achieves high performance by the use of a novel multi-level WASP module. Taking into consideration the issue of frequent occlusion in the joints of the hand, we designed HandyPose to combine feature maps from different levels of the backbone with the multi-scale approach of the WASPv2 module to obtain a more powerful representation.

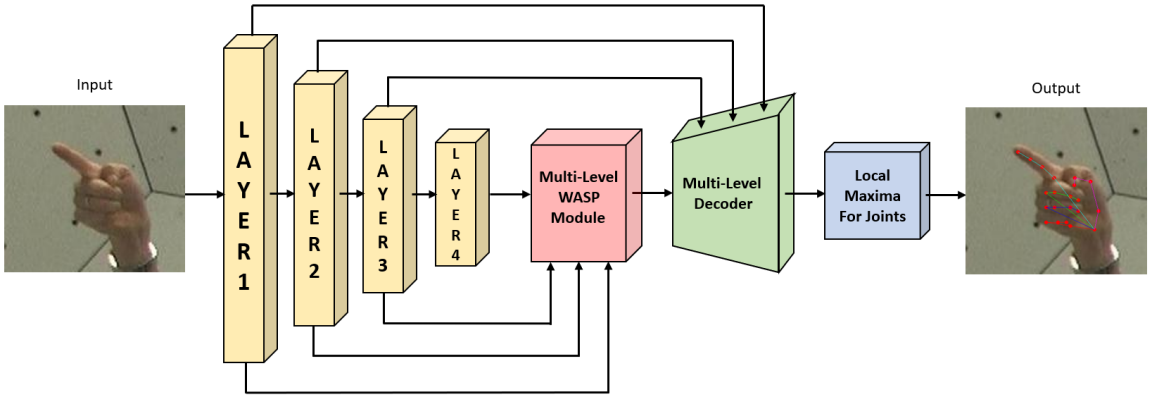

Figure 4.1: The proposed HandyPose architecture for 2D hand pose estimation. The input RGB image is fed into the ResNet-101 backbone, obtaining 400 feature maps after the concatenation of Multi-Level WASP outputs and MLF feature channnels. The Multi-Level Decoder module generates heatmaps (one per joint) and exact locations of keypoints are extracted from the heatmaps by applying a local maxima function.

The HandyPose architecture is shown in Figure 4.1. Our feature representation framework uses features from all successive blocks (levels) of the ResNet-101 backbone and incorporates them at various places in the network. Our enhanced MLW module increases the number of feature maps, forming a more robust representation, and maintains the high resolution of the maps. These representations, along with a multi-level decoder, generate more accurate predictions for both occluded and visible joints.

In order to deal with the loss of contextual and spatial information by successive pooling, our feature utilization approach fuses multi-level features extracted from all blocks of the ResNet backbone. Intermediate feature maps are fed-forward in the network with the use of a 1×1 convolution and bilinear interpolation on multi-level features to generate feature maps of matching dimensions. This helps reduce the size of the network but preserves the image information.

The High-Resolution Network (HRNet) [61] contains both high and low resolution feature maps forming a multi resolution FOV. WASPv2 combines the atrous convolutions with increasing dilation rates. HRNet and WASPv2 both benefit from the multi-scale fusion approach, by maintaining the larger FOV, a capability that we achieve in a simpler fashion with our waterfall module.

Our improved multi-level waterfall module is tailored to better extract contextual information in a challenging task with constant occlusion, such as hand pose estimation. It resembles to form a waterfall by concatenating multiple streams taken from different parts of the architecture and the waterfall module increases the network's capacity to compute multi-scale contextual information, resulting in generation of high resolution feature maps with valuable image context when compared to many generic encoder-decoder architectures.

A major challenge for CNN based architectures in both pose estimation and semantic segmentation methods is to deal with the loss of spatial information due to successive downsampling layers. HandyPose multi-level waterfall features to achieve

state-of-the-art results by fully extracting the spatial information at different levels of the backbone architecture and processing them in the WASPv2 module. HandyPose extracts four residual blocks in the ResNet backbone generating feature maps with 256, 512, 1024, and 2048 channels at all levels of the ResNet.

The MLF component of HandyPose extracts the intermediate residual feature maps at different resolutions from the first three blocks of the ResNet backbone and combines them with the high-level features of the backbone after passing through a waterfall of atrous convolutions to fuse the spatial information at different scales. We experimented with different fusing combinations and the best results were obtained by combining the multi-level feature maps in increasing order of Layer 1, 2 and 3 and concatenating them at different levels of the MLW module containing the high-level features.

Before concatenation, we perform a 1×1 convolution preceding the bilinear interpolation on the multi-level feature maps to match the resolution and shape. The unit convolutions reduce the depth channels to the desired amount of feature channels of 48 at each level. Experiments with different proportions of high and low level features of the network were performed, resulting in a higher performance when implementing 48 channels for each intermediate feature maps from the ResNet backbone to be fused with 256 feature maps from the high level features output of the ResNet backbone.

4.1.1 Multi-Level WASP Module

We present the advanced MLW module in Figure 4.3, incorporating the multi-scale extraction of the WASPv2 module shown in Figure 4.2 with the multi-level backbone features. HandyPose extracts four residual blocks in the ResNet backbone generating feature maps with 256, 512, 1024, and 2048 channels at all levels of the ResNet. We organized atrous convolutions in a waterfall like architecture receiving inputs from different levels of the ResNet backbone and concatenating them with the output

of progressive filtering of the successive layer of atrous convolutions in a waterfall fashion, as shown in Figure 4.3.

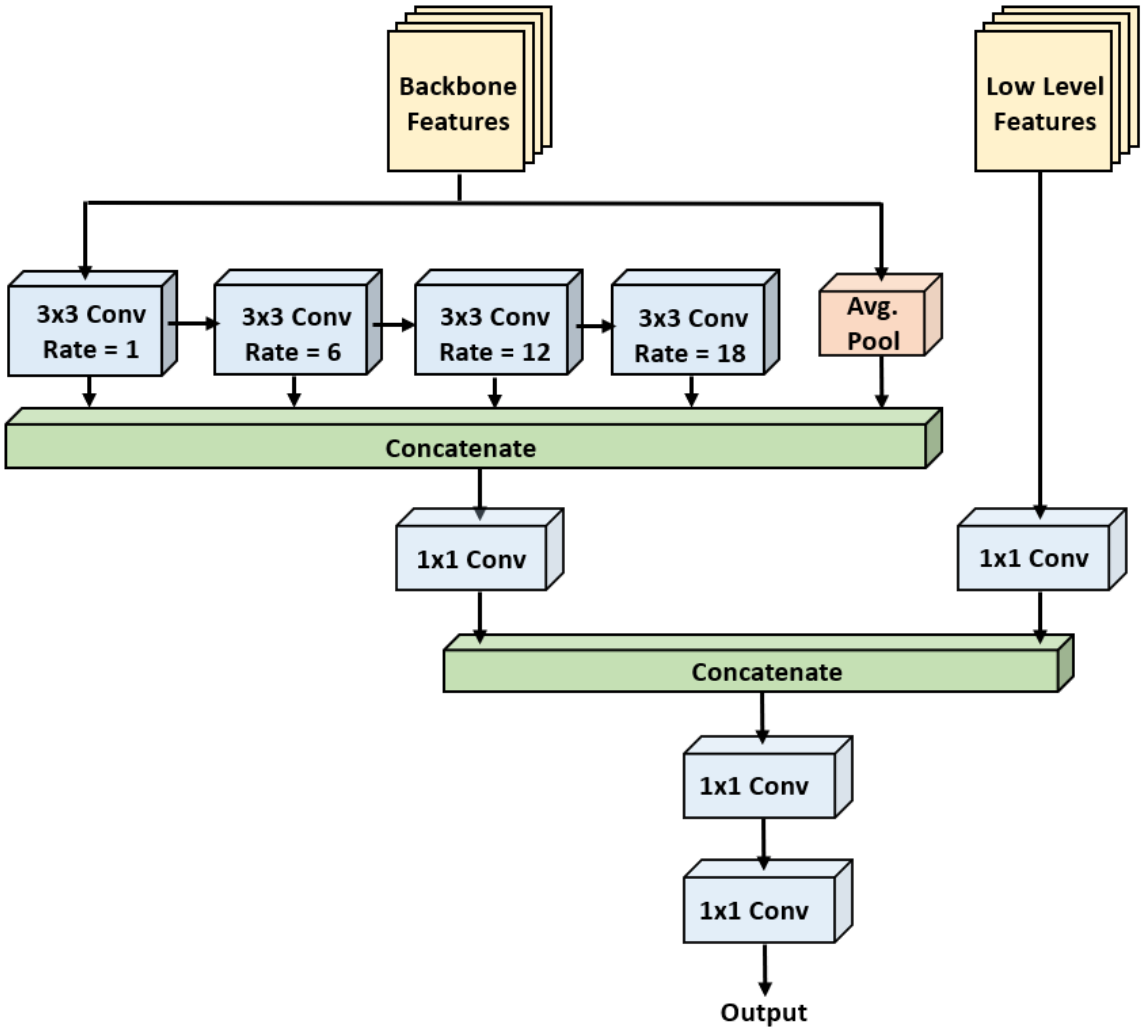

Figure 4.2: The WASPv2 waterfall module with integrated decoder. The inputs are 2048 channels of backbone features and 256 channels from the lowest and highest level of the backbone. The number of output channels is equal to the number of joints.

Our Waterfall convolutions help in maintaining the large Field-of-view (FOV) of the input features maps by avoiding downsampling and preserve the contextual information by increasing dilation rates at every step. We performed experiments with different dilation rates and the best results were obtained by progressively increasing them. We selected dilation rates of [1, 6, 12, 18] for our atrous convolutions.

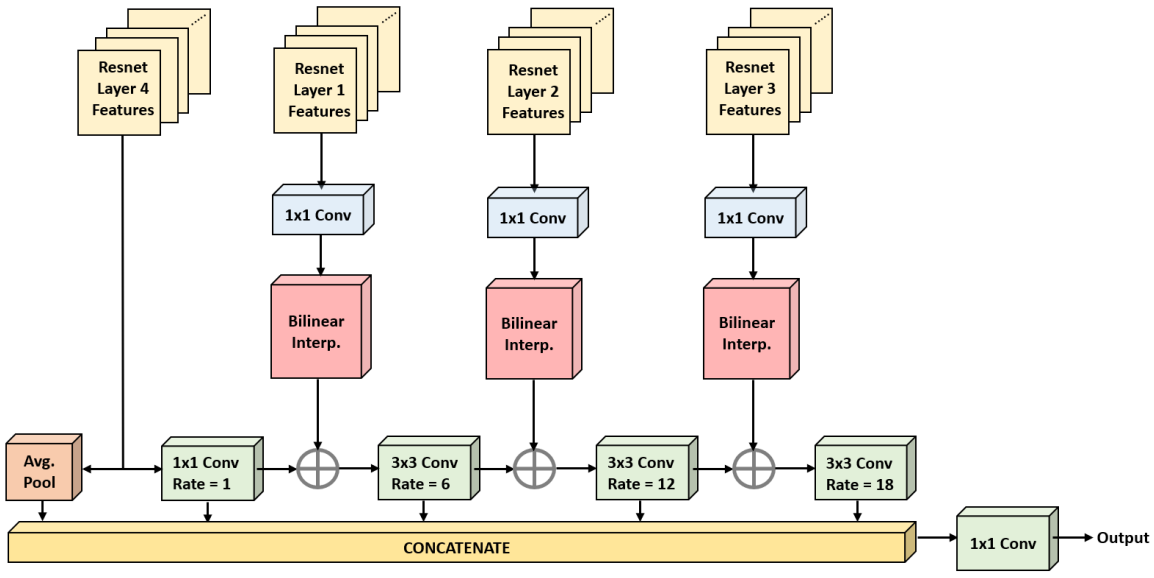

Figure 4.3: The proposed Multi-Level WASP module, a multi-level and multi-scale architecture with larger FOV for preserving the contextual information with the introduction of multi-level features along the cascade of atrous convolutions. The $\oplus$ refers to concatenation. The input is 2048 feature channels from the lowest level of the backbone and the output generates 256 feature channels that are fed into the decoder.

The operations in the MLW module can be described by the following equations:

$$F_i = \begin{cases} K_{d_1} \circledast f_4 & \text{if } i{=}1 \\ K_{d_i} \circledast (f_{i-1} \circledast K_1 + f_{i-1}) & \text{otherwise} \end{cases} \tag{4.1}$$

$$F_{Waterfall} = K_1 \circledast \Big(\sum_{i=1}^{4} (F_i) + AP(f_4) \Big) \tag{4.2}$$

where $\circledast$ represents convolution, K_1 and K_{d_i} represent convolutions of kernel size 1×1 and 3×3 with dilations of $d_i = [1, 6, 12, 18]$, f_i represents the output of block i from the ResNet backbone, and AP denotes the Average Pool operation, as shown in Figure 4.3.

Our module achieves a multi-level and multi-scale representation by fusing the cascade of atrous convolutions and MLF features and concatenating the outputs with the average pooling of the high level features from the last block of ResNet-101. The resulting feature maps are reduced in channel depth by applying a 1×1 convolution

on them. These feature maps along with the MLF are used as inputs for further processing and generating the final heatmaps in the decoder, which receives another set of intermediate level features from the backbone.

The WASPv2 and MLW modules are illustrated in Figures 4.2 and 4.3 respectively. The WASPv2 module uses only the highest and lowest level features as input followed by a cascade of atrous convolutions. In contrast, the proposed MLW module adopts a multi-level approach extracting feature maps from different levels of the network and concatenating them at different stages of the cascade of atrous convolutions. Furthermore, the WASPv2 module contains an inbuild decoder to output the final feature maps, whereas the MLW module feature maps are fed in a separate multi-level decoder module to further improve the performance.

4.1.2 Multi-Level Decoder

The feature maps generated by different stages of our HandyPose architecture are fused in the Multi-Level Decoder (MLD), generating the final K heatmaps corresponding to each hand point joint from the dataset. Figure 4.4. shows the MLD module where multi-level features are processed to generate 48 feature maps each for the first three layers of the ResNet backbone and combined with 256 score maps generated by the MLW module to form a total of 400 feature maps.

The resultant feature maps are then processed through convolutional layers and finally interpolated to generate the output heatmaps of the same size as the input images. The concatenation of multi-level features followed by the convolution and dropout layers in our MLD improves the prediction accuracy of the network by 0.8%. The multi-level features output, after concatenation with the MLW module, is described as follows:

$$F_{Concat} = AP(F_1 \circledast K_1) + \sum_{i=2}^{3}(f_i \circledast K_1) + F_{Waterfall} \tag{4.3}$$

$$F_{out} = ((F_{Concat} \circledast K_3) \circledast K_3) \circledast K_1 \tag{4.4}$$

where $\circledast$ represents convolution, K_1 and K_3 represent convolutions of kernel size 1×1 and 3×3, f_i represents the output of block i from the ResNet backbone, AP denotes the Average Pool operation, $F_{Waterfall}$ is the output from the MLW module, and F_{out} is the output of HandyPose, as shown in Figure 4.4.

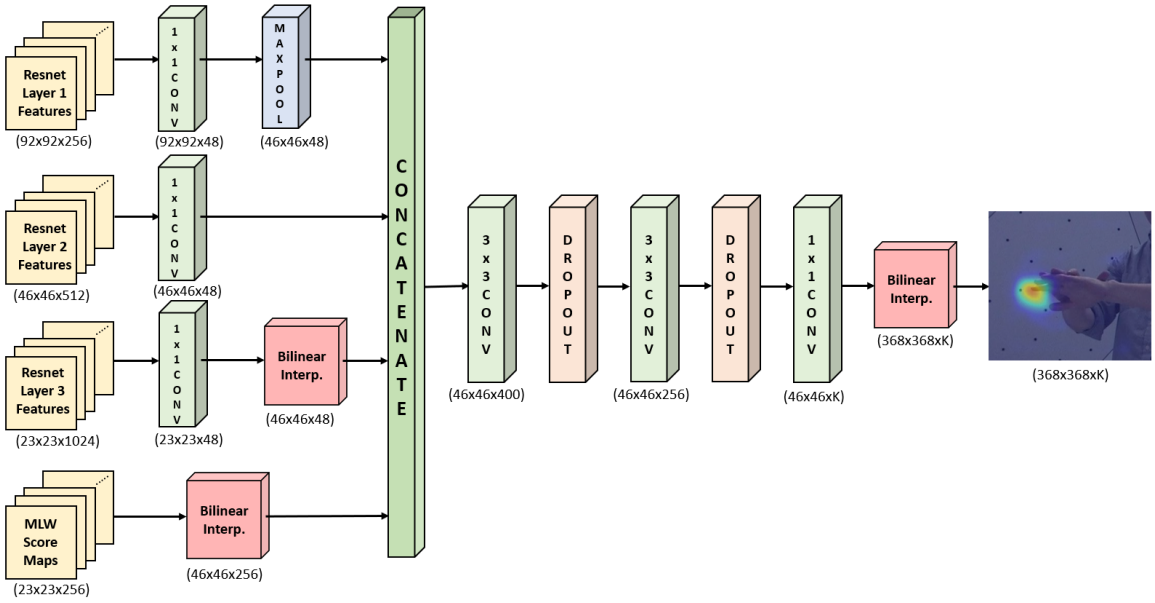

Figure 4.4: Multi-Level Decoder (MLD) module. The MLD receives [256, 512, 1024] feature maps as input from three different layers of the backbone along with the output feature maps of the MLW module. Applying, 1×1 convolution, pooling, and bilinear interpolation on multi-level feature maps results in 400 feature maps, progressively reducing the parameters of the network. Further processing through convolution and dropout layers followed by an interpolation layer generates K output heatmaps corresponding to the hand joints. The output image illustrates one channel output for HandyPose, corresponding to one joint, superimposed on the input image.

Our decoder process the output to generate heatmaps corresponding to the keypoints, extracting the joint locations through local maximum operation on each

heatmap, that is, the location in which there is the highest confidence that a joint is located. HandyPose does not require any post-processing operations as Non-Maximum Suppression (NMS) for joint localization purposes. We performed experiments with different depth channels for our multi-level feature maps and their effects on the performance.

Chapter 5

HandyPose Experiments

5.1 HandyPose Experiments

HandyPose experiments were based on metrics set by each dataset and processed at the same resolution to the dataset and other networks to allow comparison of performance.

5.1.1 Datasets

We perform 2D hand pose experiments on two hand pose datasets, the CMPU Panoptic Hand Dataset and the MPII + NZSL Dataset. Following procedures adopted by [68], [51], and [52], initial cropping of a square image patch for the annotated hands was performed in the original images, resulting in a square bounding box with dimensions 2.2× the size of the hand.

5.1.1.1 CMU Panoptic Hand Dataset

The CMU Panoptic Hand Dataset developed a multiview bootstrapping technique for generating a large annotated dataset using a weak initial detector.This dataset consists of 14,817 images taken in a panoptic studio. Each image has 21 joint annotations of a single hand and is one of the benchmark datasets for 2D hand pose estimation. The images were taken in the CMU Panoptic lab by the process of multiview bootstrapping [32].

Following procedures of other methods in the comparison section, we divided the dataset by splitting the samples into training, validation, and testing sets containing 70%, 15%, and 15% of the images, respectively. The main challenge of the dataset is the high occurrence of self-occlusion for hand keypoints.

5.1.1.2 MPII + NZSL Dataset

The MPII + NZSL dataset contains images from the MPII human pose dataset combined with the New Zealand Sign Language dataset. It consists of 2,758 images of people in everyday activities, and contains 21 labeled joint keypoints for the hand.

Following the same procedure as the previous dataset, we crop the region with the target hand. We cropped the hand region from each image by following the same procedure as we did for the CMU Panoptic Hand Dataset. Each image contains 21 joint annotations of a single hand.

5.2 Evaluation Metric

For the evaluation of HandyPose, we apply the metric of Probability of Correct Keypoint (PCK). This metric measures a correct prediction when the joint detection is within a certain distance threshold σ in relation to the hand bounding box, compared to the groundtruth label. The main threshold adopted for the dataset evaluations is $\sigma = 0.02$. Therefore, the PCK metric is defined as:

$$PCK(@.02) = P(\sigma)/K \tag{5.1}$$

$$mPCK = \frac{1}{N}\Big(\sum_{\sigma=1}^{6} PCK@[\frac{\sigma}{100}]\Big) \tag{5.2}$$

For a threshold σ of 0.02 and input image of size w $\times$ w, PCK is defined as the number of predicted keypoints (P) that are within the threshold range $\sigma \times 0.02$ of the

ground truth keypoints location divided by the total number of keypoints (k). The normalized threshold σ used is with respect to the size of the hand bounding box. We evaluated our model for different threshold values ranging from {0.01 - 0.06} with a constant increment of 0.01. We also reported the mean PCK (mPCK) showing the average performance of our network compared to current state-of-the-art methods by substituting the value of N = 6.

5.3 Implementation Details

Since the hand occupies a small area of the images in the dataset, and following procedures adopted by [68], [51], and [52], we pre-process the images, cropping it to a square with 2.2× the hand size. As the size of hand with respect to the entire image is formed by very few pixels, preprocessing of the images is required. Also, as our main focus is hand pose estimation which forms a relatively small part of the whole image, hence we cropped the images by 2.2 times the size of the square bounding box of the hand.

The cropped images are then resized to a constant resolution of 368×368, scaled between $[0, 1]$, and and normalized following procedures comparable to other methods. Resized images reduce the computational footprint of the network, and decreased the training time of the network. We used a batch size of 32 images during training, and performed training for 80 epochs, applying an initial learning rate of $lr = 10^{-4}$, being reduced by a factor of 0.1 after 60 epochs.

Chapter 6

HandyPose Results

6.1 HandyPose Results

We present HandyPose results on two prominent datasets and provide comparisons with current state-of-the-art methods. We also performed experiments by replacing our MLW module with ASPP, WASP, and WASPv2 modules. The results obtained are discussed in Section Ablations Studies.

6.1.1 Ablations Studies

We initially performed a series of experiments to analyse the accuracy, as well as computational cost and number of parameters, for each component added to our HandyPose framework. We performed a series of ablation studies to investigate the performance of atrous convolutions before developing our MLW module. Performing experiments with the ASPP, WASP and WASPv2 module on the HandyPose architecture, we observe the improvement in performance by using a cascade of atrous convolutions. WASPv2 with multi-level features is performing the best, compared to the other configurations, as shown in Table 6.1.

We also compared the use of different feature extractors and decoders for our HandyPose framework. Table 6.1 demonstrates the results for the inclusion of the ASPP module [38], WASP module [12], and WASPv2 module [39] in combination with

Method	Params (M)	GFLOPs	ASPP	WASP	WASPv2	MLW	MLD	PCK @0.2
ResNet [82]	44.6	28.3						69.20%
Deeplab [38]	59.3	34.9	✓					71.15%
Unipose [12]	47.5	29.2		✓				70.32%
WASPv2	47.0	28.8			✓			73.58%
WASPv2 + MLF	47.2	29.3				✓		73.97%
HandyPose	47.5	29.5				✓	✓	74.61%

Table 6.1: Ablation studies for different configurations of HandyPose with ResNet-101 backbone for the CMU Panoptic Hand dataset. MLW and MLD represents the Multi-Level WASP and Multi-Level Decoder modules using Multi-Level Features (MLF). ASPP, WASP, and WASPv2 indicates the use of various atrous modules in the network.

the improved feature extractor in our architecture. The combination of the modified WASPv2 module with our implementation of MLF and our MLD demonstrated to be the more efficient architecture, gaining 3.46% in accuracy (from 71.15% to 74.61%) when compared to DeepLab.

Backbone	Params (M)	GFLOPs	WASPv2	MLW	MLD	PCK @0.2
ResNet-50	25.6	19.0				65.44%
ResNet-50	27.9	19.5	✓			69.28%
ResNet-50	28.2	20.1		✓		70.13%
ResNet-50	28.5	20.2		✓	✓	70.92%
ResNet-101	44.6	28.3				69.20%
ResNet-101	47.0	28.8	✓			73.58%
ResNet-101	47.2	29.3		✓		73.97%
ResNet-101	47.5	29.5		✓	✓	74.61%
HRNet-W48	68.0	38.1				69.55%
HRNet-W48	68.2	38.9	✓			70.30%
HRNet-W48	68.2	39.3		✓		70.91%
HRNet-W48	68.3	39.6		✓	✓	71.27%

Table 6.2: Performance comparison of three different backbones, ResNet-50, ResNet-101 and HRNet-W48 in the presence or absence of different components of the HandyPose architecture for the CMU Panoptic Hand dataset.

Table 6.2 demonstrates the performance comparison of three different backbones, ResNet-50, ResNet-101 and HRNet-W48 [61], combined with different components of HandyPose on the CMU Panoptic Hand dataset. This dataset consists of images of human of resolution 1920×1080, but hands are cropped from a small part of the image due to their smaller size.

The average size of hand crops in the dataset is 44×48. From ablations performed in Table 6.2 we can infer that the multi-level feature resolutions of the ResNet backbone help to further extract important information from different scales. In comparison, the high-resolution HRNet backbone is not as effective for processing smaller portions of the image containing the hand. The ResNet-101 model improves the accuracy by 5.41% (from 69.20% to 74.61%), while the HRNet configuration improves by only 1.72% (from 69.55% to 71.27%).

Feature Maps	PCK@0.02
24	72.8%
48	**74.61%**
96	71.5%
128	70.3%

Table 6.3: HandyPose results for the CMU Panoptic Hand dataset showing the effects of varying the number of feature maps in the multi-level-features.

Table 6.3 presents a comparison for the implementation of our multi-level feature maps by applying different numbers of feature maps for the lower and intermediate blocks of the ResNet backbone into the modified MLW module and the MLD of HandyPose.

We tested our network with different numbers of feature maps for the intermediate level features $[24, 48, 96, 128]$ generated by the multi-level approach. Similar to results previously observed by architectures applying low-level features to the decoder stage [38], [12], [39], the use of 48 feature maps for lower level features and 256 maps for high level features was found to be the more efficient combination.

CMU Panoptic Hand Dataset									
Method	Params (M)	GFLOPs	PCK @0.01	PCK @0.02	PCK @0.03	PCK @0.04	PCK @0.05	PCK @0.06	mPCK
HandyPose (ours)	47.5	29.5	**43.13%**	74.61%	87.85%	92.81%	**95.28%**	**96.84%**	**81.75%**
10-head R-SiaPose-HG [68]	-	-	39.46%	**77.22%**	**88.45%**	**92.97%**	94.85%	96.09%	81.48%
UniPose [12]	47.5	29.2	36.60%	70.32%	84.81%	90.60%	93.72%	95.64%	78.61%
10-head R-SiaPose-CPM [68]	-	-	26.62%	65.80%	81.60%	88.02%	91.39%	93.36%	74.47%
R-SiaPose-CMU [68]	-	-	24.94%	62.08%	77.83%	84.91%	88.78%	91.34%	71.64%
AGMN [51]	-	-	23.90%	60.26%	76.21%	83.70%	87.72%	90.27%	70.34%
R-MGMN [52]	-	-	23.67%	60.12%	76.28%	83.14%	86.91%	89.47%	69.93%
AGMN Sep. Trained [51]	-	-	21.52%	56.73%	73.75%	82.06%	86.39%	89.10%	68.25%
CPM [11]	31.4	163.7	22.88%	58.10%	73.48%	80.45%	84.27%	86.88%	67.67%

Table 6.4: Results for 2D hand pose estimation and comparison with other state-of-the-art-methods for the CMU Panoptic Hand Dataset.

6.1.2 Experimental Results on CMU Panoptic Hand Dataset

We compared our multi-level approach to current state-of-the-art methods as shown in Table 6.4. The SiaPose method [68] considers several backbones in its configuration, including the heavyweight HG backbone. In addition, SiaPose adds up to 40% in its size by combining the backbone with the 10 heads for the refinement of predictions through graphical models. HandyPose achieved an overall best performance, with significant gains in comparison to the previous state-of-the-art while using a smaller backbone, ResNet-101. For the overall average accuracy, HandyPose achieves a mPCK of 81.75%, increasing the previous state-of-the-art. Most of the improvement of HandyPose is due to its higher capacity to precisely detect keypoints at lower thresholds, increasing the PCK@0.01 by 9.3% compared to the previous state-of-the-art (from 39.46% to 43.13%). HandyPose is an overall more accurate framework that achieves most of its gains in the fine refinement of joints detections for tight thresholds.

In contrast to other methods relying in multi-stage frameworks [68], [51], and [52], HandyPose is able to detect with higher accuracy hand joints in a single iteration network. HandyPose improves the accuracy by 6.1% to its nearest competitor for the most traditional PCK with threshold of 0.02, and an even larger 17.8% for more precise hand pose estimation in a less forgiving threshold of 0.01, attesting to the

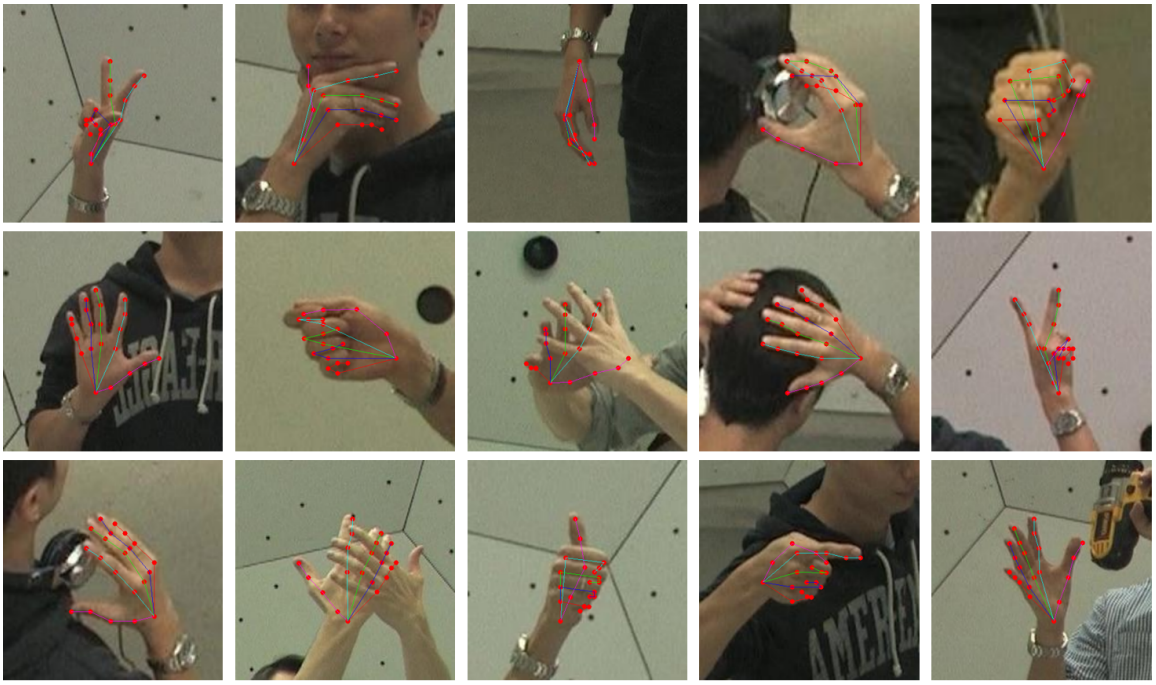

Figure 6.1: Pose estimation examples from the CMU Panoptic Hand Dataset.

more precise alignment of HandyPose to the exact joint locations.

Examples of HandyPose detections for the CMU Panoptic Hand dataset are shown in Figure 6.1. It is noticeable that HandyPose addresses with higher accuracy occluded joints, the most challenging component of hand pose estimation in general and for this dataset.

MPII + NZSL Dataset									
Method	Params (M)	GFLOPs	PCK @0.01	PCK @0.02	PCK @0.03	PCK @0.04	PCK @0.05	PCK @0.06	mPCK
HandyPose (ours)	47.5	29.5	**16.02%**	**41.66%**	**58.15%**	**68.12%**	**74.53%**	**79.90%**	**56.39%**
UniPose [12]	47.5	29.2	14.29%	38.85%	55.28%	65.14%	71.75%	77.52%	53.80%
10-head R-SiaPose-HG [68]	-	-	12.19%	33.34%	49.13%	59.86%	67.83%	73.69%	49.33%
10-head R-SiaPose-CPM [68]	-	-	8.40%	24.71%	39.33%	50.31%	59.04%	66.01%	41.30%
CPM [11]	31.4	163.7	8.05%	23.78%	37.74%	48.00%	55.65%	61.68%	39.15%

Table 6.5: Results for 2D hand pose estimation and comparison with other state-of-the-art-methods for the MPII + NZSL Dataset.

6.1.3 Experimental Results on MPII+NZSL Dataset

We next performed our experiments on the MPII+NZSL Our HandyPose framework outperformed the current SOTA methods by a significant margin as reported in Table 6.5.

Similar to results from the previous dataset, HandyPose outperforms the state-of-the-art, achieving an overall mPCK of 56.39%, increasing the accuracy from other methods by 4.8%. Significant improvements are also present for the traditional PCK with threshold of 0.02 by a margin of 7.2%, reaching 41.66%. The MPII+NZSL dataset presents more challenging images in the wild, having in addition to the high incidence of occlusion a great amount of variability of images, resulting in a more difficult dataset for architectures to predict hand pose estimation.

Figure 6.2: Pose estimation examples from the MPII+ NSZL dataset.

Examples for hand pose estimation for images from the MPII part of the dataset and the New Zealand Sign Language part of the dataset are shown in Figure 6.2 and Figure 6.3, respectively. Images from the MPII part of the dataset present a

higher challenge due to greater variation of the background in the wild, adding to the challenge of occlusion present in both parts of the dataset.

Figure 6.3: Pose estimation examples from the New Zealand Sign Language (NZSL) dataset.

Chapter 7

Conclusion

7.1 Conclusion

We presented the HandyPose and VehiPoses frameworks for 2D hand and vehicle pose estimation. VehiPose is a single-stage, end-to-end trainable framework that leverages the waterfall multi-scale approach to accurately predict the vehicle keypoints. Our approach shows promise for further use in a broader range of applications, including 3D vehicle pose estimation. Improving upon the VehiPose architecture we propose the HandyPose architecture for hand pose estimation. HandyPose is a single-stage end-to-end trainable framework that leverages multi-level and multi-scale features to more accurately predict pose estimation without losing spatial and contextual information and better addressing occlusion of keypoints. Our novel multi-level and multi-scale approach obtains state-of-the-art results on two hand pose datasets.

Hand pose estimation has drawn increasing attention during the past decade due to its similarity to full body pose estimation and usefulness in a wide range of applications including augmented reality, virtual reality, human-computer interaction, and action recognition. The high degrees of freedom in the human hand movements and frequent self-occlusion of hand joints make the task more challenging. In addition, the low resolution of the hand crops make multi-scale feature representations more challenging.

The HandyPose and VehiPose frameworks for 2D hand and vehicle pose estimation, consist of modular, end-to-end trainable networks. In HandyPose, we proposed a multi-level waterfall module and multi-level decoder to better leverage multi-level and multi-scale features and more accurately predict pose estimation without losing spatial and contextual information in the presence of occlusions of hand keypoints.

Our multi-level feature extraction approach deals more effectively with the spatial loss of resolution due to the small size of the input image and successive pooling, while achieving high accuracy and maintaining the size complexity and modularity of the network. HandyPose achieves state-of-the-art results on two hand pose datasets and sets the foundation for future work on 3D pose estimation.

Bibliography

[1] B. Artacho and A. Savakis, "Waterfall atrous spatial pooling architecture for efficient semantic segmentation," *Sensors*, vol. 19, no. 24, p. 5361, Dec 2019. [Online]. Available: http://dx.doi.org/10.3390/s19245361

[2] T. Lee and T. Hollerer, "Multithreaded hybrid feature tracking for markerless augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 355–368, 2009.

[3] Y. Jang, S.-T. Noh, H. J. Chang, T.-K. Kim, and W. Woo, "3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, pp. 501–510, 2015.

[4] S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, pp. 1–54, 2012.

[5] Y. Wang, C. Peng, and Y. Liu, "Mask-pose cascaded cnn for 2d hand pose estimation from single color image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3258–3268, 2019.

[6] N. Santavas, I. Kansizoglou, L. Bampis, E. Karakasis, and A. Gasteratos, "Attention! a lightweight 2d hand pose estimation approach," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11 488–11 496, 2021.

[7] Y. Wang, B. Zhang, and C. Peng, "Srhandnet: Real-time 2d hand pose estimation with simultaneous region localization," *IEEE Transactions on Image Processing*, vol. 29, pp. 2977–2986, 2020.

[8] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3d hand shape and pose estimation from a single rgb image," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 825–10 834.

[9] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 49–59.

[10] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *arxiv:1603.06937*, 2016.

[11] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4724–4732.

[12] B. Artacho and A. Savakis, "Unipose: Unified human pose estimation in single images and videos," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[13] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. Zhou Tianyi, and J. Yuan, "A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image," in *Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV)*, 2019.

[14] Y. Chen, Z. Tu, D. Kang, L. Bao, Y. Zhang, X. Zhe, R. Chen, and J. Yuan, "Model-based 3d hand reconstruction via self-supervised learning," *arxiv:2103.11703*, 2021.

[15] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3d training data for fine hand pose estimation," *arxiv:1605.03389*, 2016.

[16] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang, "Hand3d: Hand pose estimation using 3d neural network," *arxiv:1704.02224*, 2017.

[17] W. Chen, C. Yu, C. Tu, Z. Lyu, J. Tang, S. Ou, Y. Fu, and Z. Xue, "A survey on hand pose estimation with wearable sensors and computer-vision-based methods," *Sensors*, vol. 20, no. 4, p. 1074, Feb 2020. [Online]. Available: http://dx.doi.org/10.3390/s20041074

[18] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim, "Bighand2.2m benchmark: Hand pose dataset and state of the art analysis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2605–2613.

[19] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[20] U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz, "Hand pose estimation via latent 2.5d heatmap regression," *arxiv:1804.09534*, 2018.

[21] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," *arxiv:1803.11404*, 2018.

[22] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," *arxiv:1705.01389*, 2017.

[23] A. Boukhayma, R. de Bem, and P. H. S. Torr, "3d hand shape and pose from images in the wild," *arxiv:1902.03451*, 2019.

[24] P. Panteleris, I. Oikonomidis, and A. Argyros, "Using a single rgb frame for real time 3d hand pose estimation in the wild," *arxiv:1712.03866*, 2017.

[25] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," *arxiv:1701.01779*, 2017. [Online]. Available: https://arxiv.org/abs/1701.01779

[26] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[28] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1302–1310.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[31] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," *arxiv:1808.01244*, 2019.

[32] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[33] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua, "Direct prediction of 3d body poses from motion compensated sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[34] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Dense 3d regression for hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[35] G. Moon, J. Chang, and K. M. Lee, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[36] B. Doosti, "Hand pose estimation: A survey," *arxiv:1903.01013*, 2019.

[37] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[39] B. Artacho and A. E. Savakis, "Omnipose: A multi-scale framework for multi-person pose estimation," *arxiv:2103.10180*, 2021.

[40] H. C. Sánchez, A. H. Martínez, R. I. Gonzalo, N. H. Parra, I. P. Alonso, and D. Fernández-Llorca, "Simple baseline for vehicle pose estimation: Experimental validation," *IEEE Access*, vol. 8, pp. 132 539–132 550, 2020.

[41] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[42] S. Zhang, C. Wang, Z. He, Q. Li, X. Lin, X. Li, J. Zhang, C. Yang, and J. Li, "Vehicle global 6-DoF pose estimation under traffic surveillance camera," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 114–128, 2020.

[43] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[44] C. Keskin, F. Kiraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *ECCV*, 2012, pp. 852–863. [Online]. Available: https://doi.org/10.1007/978-3-642-33783-3_61

[45] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM Trans. Graph.*, vol. 28, no. 3, 2009. [Online]. Available: https://doi.org/10.1145/1531326.1531369

[46] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, jul 2014.

[47] D. S. Alexiadis and P. Daras, "Quaternionic signal processing techniques for automatic evaluation of dance performances from mocap data," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1391–1406, 2014.

[48] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1372–1384, 2006.

[49] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[50] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*, 2011, pp. 1297–1304.

[51] D. Kong, Y. Chen, H. Ma, X. Yan, and X. Xie, "Adaptive graphical model network for 2d handpose estimation," *arxiv:1909.08205*, 2019.

[52] D. Kong, H. Ma, Y. Chen, and X. Xie, "Rotation-invariant mixed graphical model network for 2d hand pose estimation," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 1535–1544.

[53] J. Malik, I. Abdelaziz, A. Elhayek, S. Shimada, S. A. Ali, V. Golyanik, C. Theobalt, and D. Stricker, "Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[54] A. Sinha, C. Choi, and K. Ramani, "Deephand: Robust hand pose estimation by completing a matrix imputed with deep features," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4150–4158.

[55] F. Huang, A. Zeng, M. Liu, J. Qin, and Q. Xu, "Structure-aware 3d hourglass network for hand pose estimation from single depth image," in *The British Machine Vision Conference (BMVC)*, 2018.

[56] C.-H. Yoo, S. Ji, Y.-G. Shin, S.-W. Kim, and S.-J. Ko, "Fast and accurate 3d hand pose estimation via recurrent neural network for capturing hand articulations," *IEEE Access*, vol. 8, pp. 114 010–114 019, 2020.

[57] A. Pemasiri, K. Nguyen, S. Sridharan, and C. Fookes, "Unified 2d and 3d hand pose estimation from a single visible or x-ray image," in *BMVC*, 2019.

[58] J. Tompson, M. Stein, Y. LeCun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, pp. 169:1–169:10, 2014.

[59] B. Liu, S. Huang, and Z. Ye, "Networks effectively utilizing 2d spatial information for accurate 3d hand pose estimation," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 574–578.

[60] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[61] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," *arxiv:1902.09212*, 2019.

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is all you need," ser. NIPS'17. Curran Associates Inc., 2017, p. 6000–6010.

[63] Y. Chen, H. Ma, D. Kong, X. Yan, J. Wu, W. Fan, and X. Xie, "Nonparametric structure regularization machine for 2d hand pose estimation," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 370–379.

[64] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," 2019.

[65] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," 2020.

[66] J. Song, L. Wang, L. Van Gool, and O. Hilliges, "Thin-slicing network: A deep structured model for pose estimation in videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5563–5572.

[67] X. Chen and A. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 1736–1744.

[68] D. Kong, H. Ma, and X. Xie, "Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation," *arxiv:2009.12473*, 2020.

[69] B. Doosti, S. Naha, M. Mirbagheri, and D. J. Crandall, "Hope-net: A graph-based model for hand-object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[70] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[71] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *arxiv:1505.04366*, 2015.

[72] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, 2016.

[73] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arxiv:1706.05587*, 2017.

[74] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, p. 652–662, Feb 2021. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2019.2938758

[75] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," 2019.

[76] R. Juránek, A. Herout, M. Dubská, and P. Zemcík, "Real-time pose estimation piggybacked on object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2381–2389.

[77] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," *arxiv:1804.02767*, 2018.

[78] W. Ding, S. Li, G. Zhang, X. Lei, and H. Qian, "Vehicle pose and shape estimation through multiple monocular vision," *arxiv:1802.03515*, 2018.

[79] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Occlusion-net: 2D/3D occluded keypoint localization using graph networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7326–7335.

[80] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2011–2018.

[81] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arxiv:1411.4038*, 2015.

[82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[83] Z. Wang, L. Tang, X. Liu, Z. Yao, S. Yi, J. Shao, J. Yan, S. Wang, H. Li, and X. Wang, "Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 379–387.